Connection of a historical social network with Wikimedia community

Rafael Díaz Rodríguez

Abstract—Researchers from the Computer Vision Center and the Center for Demographic Studies of the Autonomous University of Barcelona are developing a research project with the aim of building a historical social network (XARXES) gathering local censuses from diverse Catalonia cities. There is a growing interest in having this data fed with as much information available as possible. The aim of this final degree project is to understand how the XARXES project and Wikipedia work, and enlarge the historical social network information by searching and integrating appropriate data from Wikipedia. Wikipedia has been chosen because it is one of the most, if not the most, important shared knowledge database in the world. The XARXES project has also a web page in development to be able to access all this censuses data. The objectives for this final degree project are two: developing a way of connecting the XARXES web page with Wikipedia and designing and developing an appropriate mobile app that works as a companion for the full web page.

Index Terms—Angular, Demographic Past, Ionic Framework, TypeScript, Wikidata, Wikipedia

1 Introduction

THIS project is part of a bigger research project called XARXES [1] and aims to make a connection between it and Wikipedia [2]. For this reason, it is important to first understand the nature of Wikipedia, Wikidata [3] and the XARXES project.

XARXES is a project in development by the CVC [4] and the CED [5] that is collecting historical censuses from Catalonia cities. They are using computer vision to automatize the handwriting recognition of the censuses and making a record linkage between all the sources to build the social network. The information gathered usually includes date of birth, the city where the census was taken, if the person was married, the place where the person was living and occupation. This information is interesting but lacks context. If you are not familiar with the events that happened in 1900, you may not be able to understand why this people moved to another city or changed jobs. That is why implementing the big knowledge of Wikipedia is an important and logic step for this project.

Wikimedia Foundation [6] is the home of many organizations, including Wikipedia, Wikidata, MediaWiki [7] or Wikiquote [8]. In this final degree project we are going to consider the first two. Wikipedia is widely known as an encyclopedia maintained by shared knowledge. They focus on having references on each article to keep it safe and reliable and nowadays contains more than 300 available languages, about 48 million available articles with five million written in English [9]. The numbers are outstanding and, to maintain all these articles, as they usually share the same information between different languages, Wikimedia community came with the idea of Wikidata: a connected source of properties and items that feeds some of Wikipedia articles information. Wikimedia Foundation is very active and is everyday looking for new opportunities to bring knowledge to everybody.

- *Contact email: rafael.diazro@e-campus.uab.cat*
- *Branch: Software Engineering*
- *Tutors: Alicia Fornés Bisquerra (Computer's Science Department), Joana Maria Pujadas Mora (Center for Demographic Studies)*
- *2017/18 course*

2 State of the Art

There is a growing base of shared knowledge websites. There are ones that are question focused like Quora [10] or Stack Exchange [11] (home of Stack Overflow [11]), and others are encyclopedia focused, like Wikipedia or Wikia [12]. Wikimedia Foundation is stronger than ever with nearly three thousand active users that makes at least one edit per month and is the fifth most used website [13]. For this reason, in this final degree project we are going to use Wikipedia as a source of knowledge. On the other side, the XARXES project is using ground-breaking computer vision and record linkage to construct a database based on historical censuses. Regarding the mobile app panorama, a new trend is growing: using the power of web technologies to build multi platform apps thanks to the use of frameworks. In this scenario, Ionic [14], Flutter [15], and React Native [16] are the most supported tools. In this project I will use the Ionic Framework because it provides an easy to use CLI and is the oldest one, so it has a big community behind it.

3 Objectives

This project has two objectives: connecting the XARXES project with Wikipedia and enriching the information provided and also developing a mobile app that is user-friendly and intuitive.

3.1 Sub-objectives

- Designing a mobile app user interface.
- Testing the user interface.

- Understanding the data that the historical social network database provides.
- Understanding the data structure that Wikidata uses in order to be able to make queries.
- Using or creating a mock database to be able to develop the app before having access to the XARXES API.
- Connecting the app to the XARXES API once it is accessible.
- Code testing.

4 Methodology

4.1 Agile

The project is composed by eight two-week iterations. Each iteration has concrete objectives and a deadline. If the objectives are not met, they are moved to the next iteration gaining more priority.

4.2 Rapid Prototyping

The UX design is developed doing first a low-fidelity prototype (or wireframes). After the design receives feedback from UX experts, the high-fidelity prototype, with UI interactions, is made. This is done using Sketch [17] and Marvel App [18], providing an interactive design that allows to make a valid testing prototype to receive feedback from potential final users and improve the final coded design.

4.3 Tasks Prioritization

There is a need to keep track the status of pending, in progress and done tasks. This is done thanks to Trello [19]. The tasks are divided in three columns depending on the status and each one has a priority label assigned (low, mid or high). The labels have colors (green, yellow or red) to identify them easily, as you can see in Fig. 1.

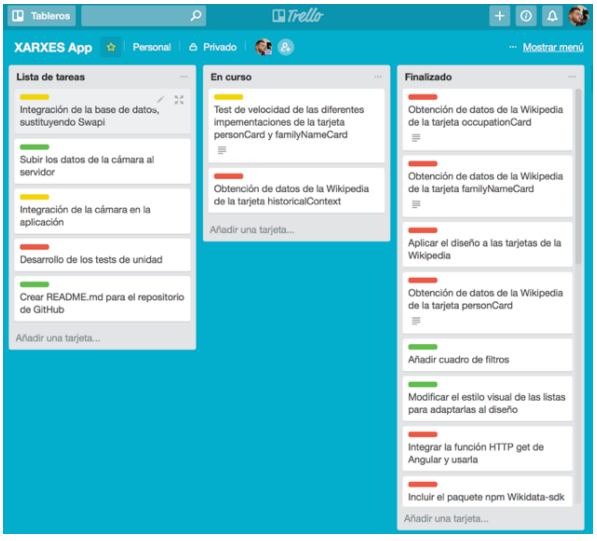


Fig. 1. Using Trello to have a track of tasks, in which is easy to differentiate the priority and status of each task.

4.4 Git Flow

Following Vincent Driessen development model [20], the mobile app code is using Git [21] and consists of two main branches: develop and master.

At the master branch there is only final releases of the app, accessible to everyone. This branch is also connected to the Ionic Framework, so it builds and compiles the latest deliverable app automatically. The releases are also tagged following the convention vX.Y.Z, where 'X' means the major release version, 'Y' determines a feature being added and Z a minor release where bugs are solved. Master can branch into hotfix/X.Y.Z to solve any detected bug. This is known as Semantic Versioning [22].

At develop branch is where all the development occurs and branches into feature/X.Y.Z where a feature can be developed or into release/X.Y.Z, where the code is prepared to merge into master for a release.

All branches are cleaned once merged, leaving a clean Git tree.

5 Requirements

5.1 Functional

Table 1 shows functional requirements for this project.

TABLE 1
Functional Requirements

Id	Detail
FR1	As soon as the user opens the app, the app will show three tabs, the search tab, the favorites tab and the info tab, and the search tab will be the active.
FR2	If a user performs a search, the app will show all the results that contain that particular name and display the name, the date of birth, the date of death if available, and the city in which the person name appears.
FR3	If a user clicks on a person from the list of results from the search, the app will show a details page containing details about that person, which includes: full name, date of birth, date of death if available, a photo if available, a link to the web page, the original censuses, the related people, a favorite button and the available Wikipedia cards.
FR4	When the user clicks on the favorite button, the person will be saved in a separate list that is accessible via the favorites tab.
FR5	As soon as the person detail page is loaded, the available Wikipedia cards will be show, which includes person, family name, occupation and historical context information.

5.2 Non-Functional

Table 2 shows the non-functional requirements for this project.

TABLE 2
NON-FUNCTIONAL REQUIREMENTS

Id	Detail
NFR1	The search page of the app should be composed of an input text box, search button and a filter button.
NFR2	The app should be easy and understandable without further explanation.
NFR3	The app should have a color scheme that mimics or follows UAB colors.
NFR4	The app will be able to access the data from the XARXES project via an API.
NFR5	The app should be compatible with iOS and Android devices.
NFR6	The app will use Wikipedia cards that provide a connection to Wikipedia.
NFR7	Wikipedia cards should also be accessible from the web page.
NFR8	The app should have a fast response, no more than two seconds of waiting for each interaction.
NFR9	Wikipedia cards should load in less than one second.

6 PLANIFICATION

6.1 Calendar

As the project has deadlines, planification is an important part. Together with Agile and the planned iterations there is a calendar made so it is easy to know the status in which the project should be, along with the deadlines.

6.2 Gantt Diagram

To have a broader vision of the time needed to develop the project, there is a Gantt Diagram [23] that can be found in Appendix A. This has three main tasks: App UI Design, Wikipedia Information Cards and XARXES integration. Each one has subtasks.

7 SYSTEM DESIGN

7.1 Use Case

For the retrieval of data from the historical social network, there is an API being developed.

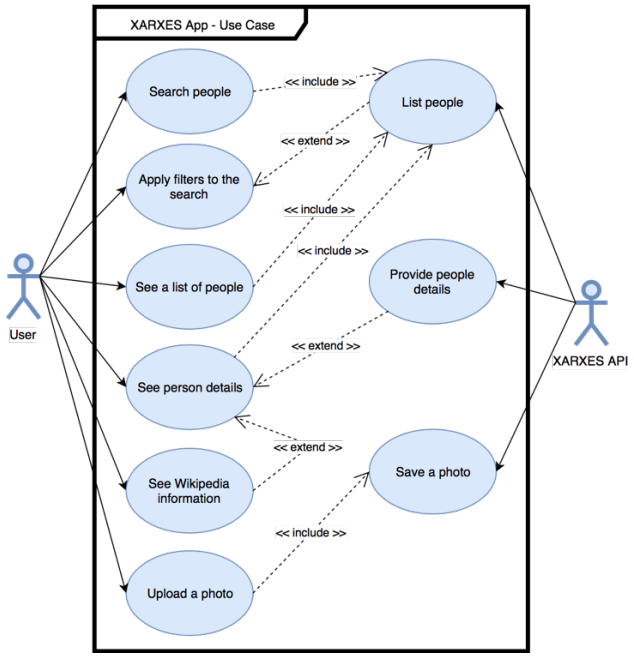


Fig. 2. Use Case Diagram involving a User and the API.

This API can provide search results from a person's name (which already includes some basic information about the person), give more detailed information about a particular person or save a photo to the server and link it to a person.

As seen in Fig. 2, the user can perform a search, with or without filters, see a list of results, see the details of a particular person, which can have Wikipedia information or not and upload a photo from the person in case this does not have any.

8 UX/UI DESIGN

8.1 Low-fidelity Prototype

This very first design consists on drawing without giving importance in how it visually looks (Fig. 3). This means that the colors are not decided yet nor applied. The most important thing is to decide the main structure and the user flow.

Taking into account the use case (7.1) this results in a user focused app, which is divided into three main pages controlled by tabs: the search page, the favorites page and the information page.

The first thing that the user sees is the search page. Here, the user can see a big search box and a big search button. The user can enter any combination of name, second name or both name and second name and easily proceed to make a search. A filter box is also shown, which expands to discover a form that lets the user edit the most usual aspects of the search.

In the favorites page the user can see all the favorited people and in the information page there is a text explaining the XARXES project and a link to the official page.

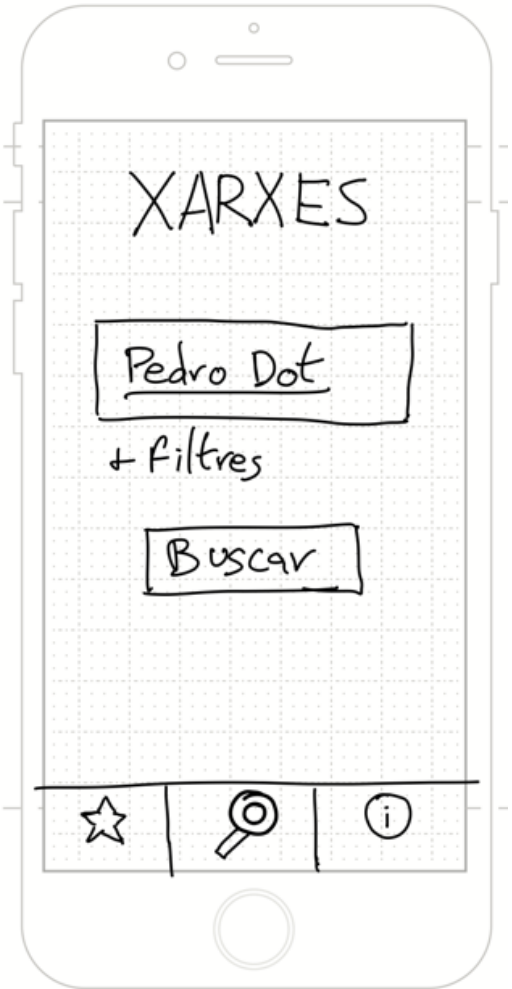


Fig. 3. Low-fidelity prototype of the search page. Notice the tabs at the bottom and the lack of color.

If the user performs a search, the list page is shown, where there is list of the applied filters and a list of the people with basic information: name, the city from where there is census information about the person and birth and death date.

When the user chooses a person from the list, the details page is shown. Here there is block of information from the person that is shown in the project web page. There is also a part where Wikipedia articles cards are displayed.

8.2 High-fidelity Prototype

This type of prototype previews the design in detail. This is done using Sketch. The prototyping app permits the export of each view (artboards) to Marvel App, which allows to create an interactive design for further user testing. Sketch also allows to copy the CSS style of each component, making it faster to match the code components with the design.
Comparing the Fig. 3 with Fig. 4 it is easy to see the differences between both prototypes.

The color decision for this design is based on green due to UAB and has an orange color which allows to make a high contrast on important components.

Fig. 4. High-fidelity prototype of the search page. Notice the definition of all the components and the presence of color.

8.3 Wikipedia Cards

Showing the information retrieved from Wikipedia is not as trivial as it seems. There are two main approaches: integrate the information of Wikipedia article into the person detail's page or showing links to Wikipedia. As there is a big amount of information available and integrating a part of a Wikipedia article could make the experience of reading information too difficult, the approach for this problem is to show enriched links to Wikipedia articles.

The information needed is the article's header, description and link. As the design intention is to make this visually appealing, the design considers also an image, retrieved from the article if it is available, or showing a default image if not.

Fig. 5. Design of Wikipedia cards. This one shows the information found on Wikipedia of a particular person.

The design takes into account that these Wikipedia cards need to be shown both on a mobile app and in a desktop web page, so it takes advantage of the space, overlapping text on the image, which is the main visual component of the card. At Fig. 5 you can see the design of a Wikipedia card showing information of a person, in this case, Pedro Dot, who is a famous Spanish rose breeder.

9 Code Design

9.1 Tools

There are some tools used to develop this project that are worth mentioning:

- The Ionic Framework is a free and Open Source project, with a wide community that has been using it for years. It is built on top of Angular [24], so it uses TypeScript [25], a typed superset of JavaScript and HTML5 and SASS [26], a CSS extension language. This framework allows to develop an app using web technologies. The code can be exported to an Android Studio [27] or Xcode [28] project to have a final built sent to the Google Play or App Store. There is also a CLI to interact with the framework that helps in creating and exporting projects.
- Visual Studio Code [29] is a full featured, lightweight and Open Source IDE from Microsoft. It includes full support for TypeScript and has a built-in terminal, which is good to interact with Ionic CLI while developing.
- Chrome [30] is a well-known web browser from Google. It features rich web developer tools that are valuable for any web development process.

9.2 Wikidata SDK

One of the functions of Wikidata is to serve information to Wikipedia. This is why, for example, if a famous person dies, the information is only updated once, in Wikidata, and is spread over all Wikipedia articles that refer to that person. The most important pieces to understand Wikidata are items [31] and properties [32]. Items always start with the letter 'Q' followed by number and refers to a real-world object, concept, or event. The item Q2 stands for Earth, for example, and item Q34981 stands for Isaac Asimov. A property (or attribute) is a descriptor for a data value within an item and always start with a 'P' followed by a number. Tim Berners-Lee (Q80) has, for example, the property occupation (P106) that contains computer scientist (Q82594).

In order to access this data, Wikidata [33] has its own API. Besides this, accessing it is a tough process and the data returned is nested in a JSON with information that may not be needed.

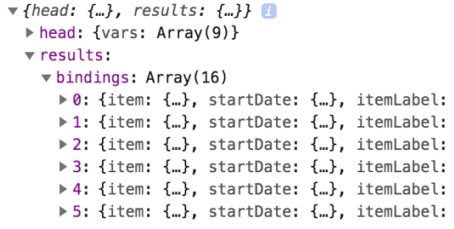


Fig. 6. Results from a Wikidata query without being simplified.

For this reason, this project uses a npm package called wikidata-sdk [34] from Maxime Lathuilière (also known as maxlath). It is able to query Wikidata in an easier way and also is able to simplify the results. It also comes with handy tools, such as a time convertor that adapts the date from Wikidata format to a more common usage format. Comparing Fig. 6 with Fig. 7 you can see the difference between the original JSON and the simplified one.

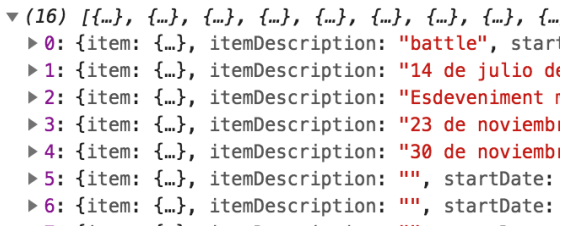


Fig. 7. Results from a Wikidata query simplified using Wikidata SDK.

9.3 Wikidata Connection Code Structure

Having the tools listed in 9.1, there are two ways of performing a search in Wikidata: searching entities using Wikidata API or making a Wikidata query using the SPARQL language [35].

In this project, both methodologies are used. Searching entities is a good way to match people. Famous people appear easily on the top results, so making a search of a full person name is a good way to find the entity. On the other side, finding an occupation or a family name is a hard task if it is done looking for entities because it can return results that are not filtered and that could be any topic. For these cases, a SPARQL is better for walking the graphs and apply restrictions.

Knowing this, in Fig. 9 you can see how the code of Wikipedia person card is structured. The utility Wikidata SDK is abbreviated as 'wdk'. The function searchEntities gets the name of the person as a parameter. This returns a URL that contains a JSON with only one result and is downloaded using an HTTP GET. From this JSON, Wikidata entity is retrieved and using getEntity function, the data that we are looking for can be extracted. Before that, it uses the helper 'simplify' as explained in 9.2.

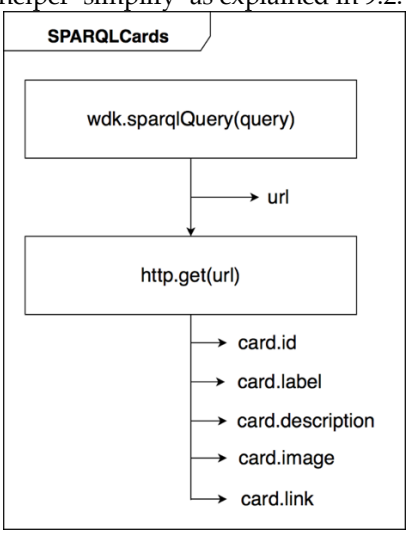


Fig. 8. Code structure for the other Wikipedia cards: occupation, family name and historical context.

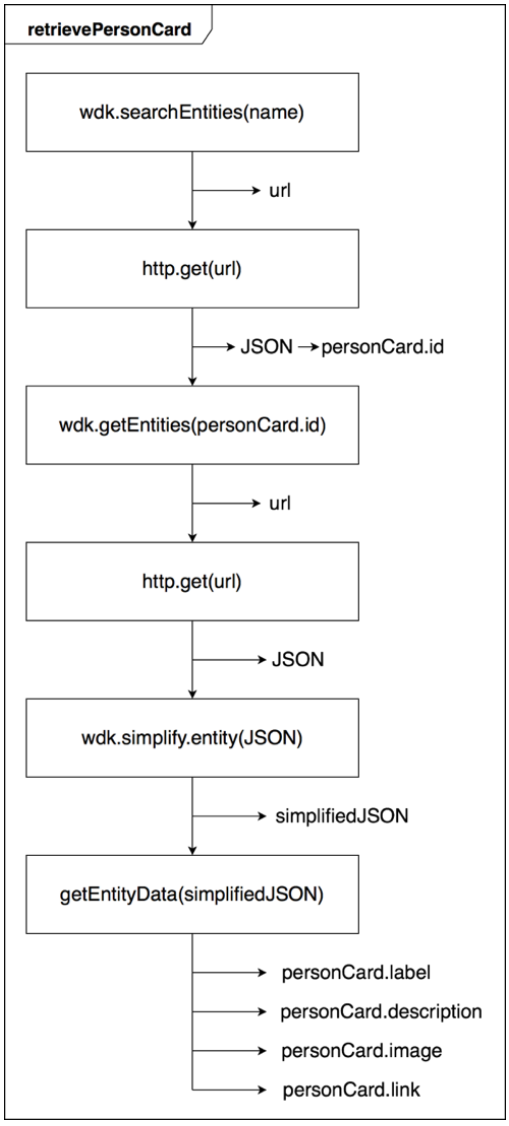


Fig. 9. Code structure for Wikipedia person card.

The occupation, family name and historical context card use the code structure from Fig. 8. It is, as seen, simpler: it uses the sparqlQuery function that already returns a JSON with the results. This is because at the SPARQL query the information needed is already filtered and categorized. At Fig. 10 you can see an example of the code of an SPARQL query.

```
SELECT ?item ?itemLabel ?itemDescription ?image
    ?articleEN ?articleES ?articleCA
WHERE {
  ?item wdt:P31 wd:Q28640.
  ?item ?label "${occupation}"@es.
  OPTIONAL { ?item wdt:P18 ?image }
  OPTIONAL { ?articleEN schema:about ?item .
    ?articleEN schema:isPartOf <https://en.wikipedia.org/>. }
  OPTIONAL { ?articleES schema:about ?item .
    ?articleES schema:isPartOf <https://es.wikipedia.org/>. }
  OPTIONAL { ?articleCA schema:about ?item .
    ?articleCA schema:isPartOf <https://ca.wikipedia.org/>. }
  SERVICE wikibase:label {
    bd:serviceParam wikibase:language "${fallback}". }
} LIMIT 1
```

Fig. 10. SPARQL query code that looks for a particular occupation on Wikidata. Notice the information that returns in the first two lines.

With this code and the UI design in mind, the visual result of the mobile app is like the one seen in Fig. 11, which shows the results for a person from the XARXES network with the discovered Wikipedia information.

Fig. 11. Historical context cards shown within a person search in the XARXES app.

10 Testing

10.1 Wikipedia Cards Performance

While developing the cards it is important to keep in mind performance. SPARQL queries can be a bottleneck in performance depending on what entities they look for. If a query is made looking for the entity people, as Wikidata contains big amounts of information for this entity, it will take seconds or minutes to charge.

To solve this issue, the query can be limited. Querying for Spanish people reduce the time considerably. Making a SPARQL query for person still takes minutes despite using restrictions, which enforces the idea to use the search entities function.

Despite that, occupation, family name and historical context are retrieved with a query. To check if the time for them was too high, the JavaScript function performance.now( ) [36] was used. Table 3 shows the results for each card, being executed 4 times.

It is important to know that browsers have a restriction when loading scripts at the same time, so this result can be affected by that. Besides that, the result shows that the person card is still affected by the downsides of having two consecutive HTTP GET. As it has to be synchronous, the waiting time makes it slower.

TABLE 3
PERFORMANCE OF EACH CARD (MS)

Person	Occupation	Family Name	Historical Context
431.5	25.0	22.8	45.9
522.5	20.6	432.0	33.4
647.3	25.2	393.0	29.1
428.3	47.2	31.2	371.8

10.2 UI Prototype Testing

The UI prototype was tested on two different people profiles: one that is expert on the computer field and one that is not. The testing was, for both, done in the same way, following the book Rocket Surgery Made Easy [37] tips and tricks.

One of these tricks is informing the tester of the purpose of the testing activity, that is merely to improve the app and knowing what is good and what is not about the app. It is important to explain that the tester is not evaluated and that there is no wrong answer. After a brief introduction, they were guided through with a few instructions together with open questions like: what do you think this is for? At the end of the activity, the person was rewarded with candies and thanked. The custom template I build to follow with each testing interview can be found in Appendix B.

The results, despite the different profiles, were surprisingly the same. There were complaints about the default image shown in cards, that was confusing, the subject of Wikipedia cards, which was not clear, readability issues and confusing logos. The problem description and the solution for each problem can be seen in Appendix C.

11 CONCLUSION

This final degree project had two objectives: connecting the XARXES project with Wikipedia and enriching the information provided and also developing a mobile app that is user-friendly and intuitive. Both have been achieved. While developing the connection with Wikipedia, Wikimedia Hackathon [38] took place in Barcelona and we were there, learning how to approach this idea. The main code for Wikipedia person card was develop there and the result was shown in a brief showcase from the attendees.

11.1 UI Design

One of the functions Design takes an important part in the experience that the user has. Without the further prototype testing, the design can fall into a failed design. This is due to biased opinions and one-sided views. The designer needs to prove the design to different types of people.

The test here has proven to be a solid help and has taken the design to the level where it really approaches the intention of the app: being easy, fast and understandable.

11.2 Wikimedia Connection

Knowing how Wikidata works is an important part to be able to develop project. At first, it was confusing, and SPARQL and the API were not friendly at first. Luckily there are tools built by the community that helps, like Wikidata Query Service [39].

Discovering the community and the values that are behind it are also worth mention. They, with their enthusiasm, are building the Internet and establishing the base by giving and sharing knowledge.

Integrating Wikipedia into XARXES is possible and allows us to have a broader and more interesting view of the data, enriching it.

This integration also opens the possibilities and allows us to think in a greater integration, like enriching Wikidata with the XARXES database.

ACKNOWLEDGMENT

Thanks to Alicia Fornés and Joana Maria Pujadas for their time, help and ideas. Thanks to Toni Hermoso for being always supportive and helpful. He encouraged us to assist to Wikimedia Hackathon of 2018, which took place in Barcelona. There we learned valuable things and met incredible people, like Maxime Lathuilière, who we also thank and that explained us and directly helped in the usage of Wikidata SDK tool. Thanks to Mario Arias, who helped with his JavaScript knowledge. Also, special mention and thanks to the person that has helped with the design and has been the greatest support of all, at all times, Janira Gilarte.

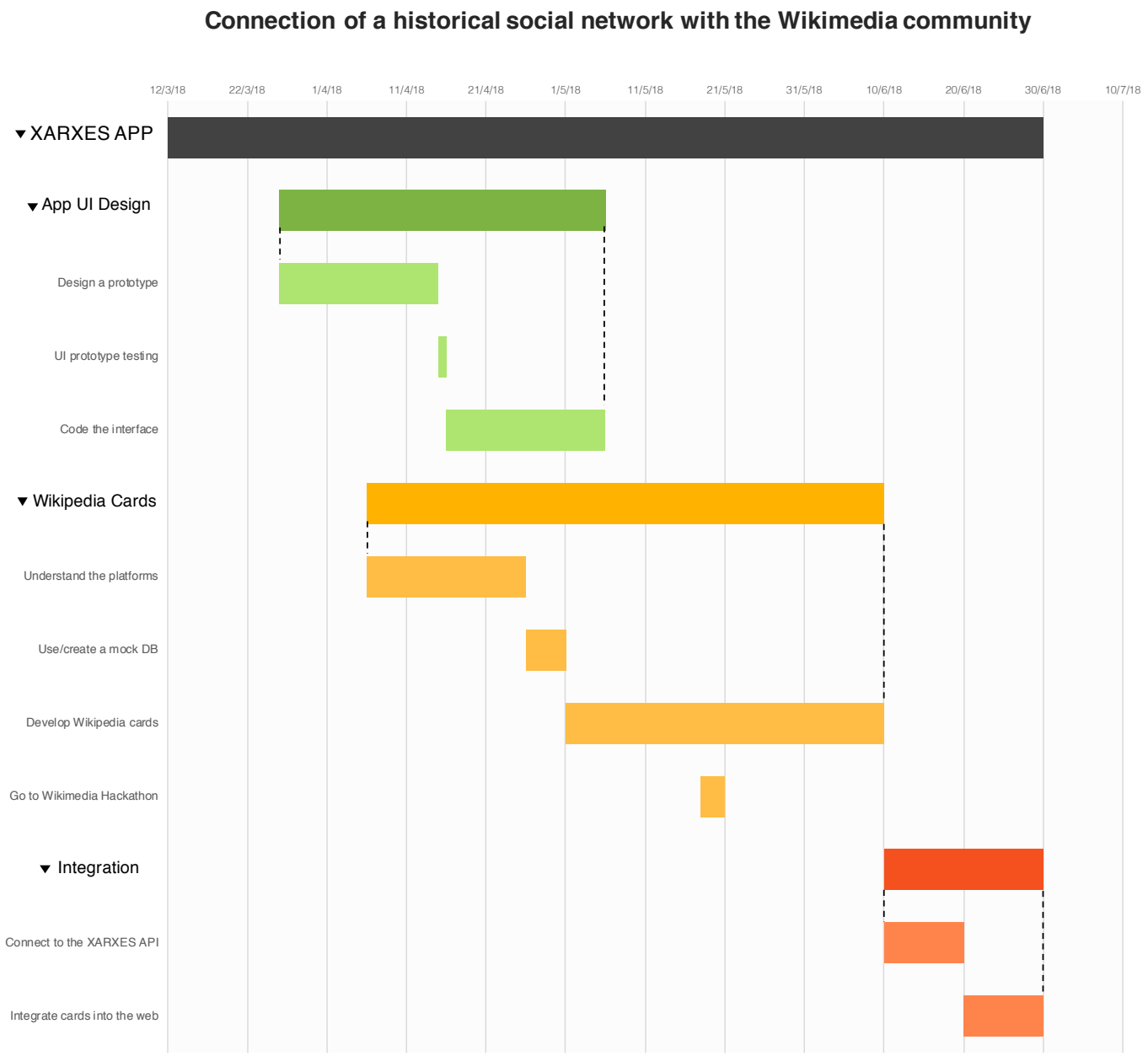
REFERENCES

[1] Computer Vision Centre, "INFO | XARXES." [Online]. Available: http://dag.cvc.uab.es/xarxes/info/. [Accessed: 09-Mar-2018].
[2] Wikimedia Foundation, "Wikipedia, the free encyclopedia." [Online]. Available: https://en.wikipedia.org/wiki/Main_Page. [Accessed: 01-Jul-2018].
[3] Wikimedia Foundation, "Wikidata." [Online]. Available: https://www.wikidata.org/wiki/Wikidata:Main_Page. [Accessed: 01-Jul-2018].
[4] "Computer Vision Center." [Online]. Available: http://www.cvc.uab.es/. [Accessed: 01-Jul-2018].
[5] "Centre d'Estudis Demogràfics." [Online]. Available: http://ced.uab.es/sobre-el-ced/benvinguda/. [Accessed: 01-Jul-2018].
[6] "Wikimedia Foundation." [Online]. Available: https://wikimediafoundation.org/wiki/Home. [Accessed: 01-Jul-2018].
[7] Wikimedia Foundation, "MediaWiki." [Online]. Available: https://www.mediawiki.org/wiki/MediaWiki. [Accessed: 01-Jul-2018].
[8] Wikimedia Foundation, "Wikiquote." [Online]. Available: https://en.wikiquote.org/wiki/Main_Page. [Accessed: 01-Jul-2018].
[9] "List of Wikipedias." [Online]. Available: https://en.wikipedia.org/wiki/List_of_Wikipedias.
[10] "Quora." [Online]. Available: https://es.quora.com/about. [Accessed: 01-Jul-2018].
[11] "Stack Exchange." [Online]. Available: https://stackexchange.com/. [Accessed: 01-Jul-2018].
[12] "Wikia." [Online]. Available: http://www.wikia.com/explore-es. [Accessed: 01-Jul-2018].
[13] Amazon, "Alexa Top 500 Global Sites." [Online]. Available: https://www.alexa.com/topsites. [Accessed: 01-Jul-2018].

[14] Ionic, "Build Amazing Native Apps and Progressive Web Apps with Ionic Framework and Angular." [Online]. Available: https://ionicframework.com. [Accessed: 09-Mar-2018].
[15] "Flutter - Beautiful native apps in record time." [Online]. Available: https://flutter.io/. [Accessed: 01-Jul-2018].
[16] "React Native." [Online]. Available: http://reactnative.com/. [Accessed: 01-Jul-2018].
[17] Sketch, "Sketch - The digital design toolkit." .
[18] Marvel, "Free mobile & web prototyping (iOS, iPhone, Android) for designers – Marvel." .
[19] Atlassian, "Trello." [Online]. Available: https://trello.com/home. [Accessed: 09-Mar-2018].
[20] V. Driessen, "A successful Git branching model." [Online]. Available: https://nvie.com/posts/a-successful-git-branching-model/. [Accessed: 01-Jul-2018].
[21] "About - Git." [Online]. Available: https://git-scm.com/about. [Accessed: 01-Jul-2018].
[22] T. Preston-Werner, "Semantic Versioning." [Online]. Available: https://semver.org/. [Accessed: 01-Jul-2018].
[23] "What is a Gantt Chart?" [Online]. Available: http://www.gantt.com/. [Accessed: 01-Jul-2018].
[24] Google, "AngularJS." [Online]. Available: https://angularjs.org/. [Accessed: 01-Jul-2018].
[25] Microsoft, "TypeScript - JavaScript that scales." [Online]. Available: http://www.typescriptlang.org/. [Accessed: 01-Jul-2018].
[26] H. Catlin, N. Weizenbaum, and C. Eppstein, "Sass: Syntactically Awesome Style Sheets." [Online]. Available: https://sass-lang.com/. [Accessed: 01-Jul-2018].
[27] Google, "Cómo descargar Android Studio y SDK Tools | Android Studio." [Online]. Available: https://developer.android.com/studio/index.html. [Accessed: 09-Mar-2018].
[28] Apple, "Xcode - Apple Developer." [Online]. Available: https://developer.apple.com/xcode/. [Accessed: 09-Mar-2018].
[29] Microsoft, "Visual Studio Code - Code Editing. Redefined." [Online]. Available: https://code.visualstudio.com/. [Accessed: 09-Mar-2018].
[30] Google, "Chrome." [Online]. Available: https://www.google.com/chrome/. [Accessed: 01-Jul-2018].
[31] "Help:Items - Wikidata." [Online]. Available: https://www.wikidata.org/wiki/Help:Items. [Accessed: 01-Jul-2018].
[32] "Help:Properties - Wikidata." [Online]. Available: https://www.wikidata.org/wiki/Help:Properties. [Accessed: 01-Jul-2018].
[33] "Wikidata API help." [Online]. Available: https://www.wikidata.org/w/api.php. [Accessed: 01-Jul-2018].
[34] M. Lathuilière, "wikidata-sdk - npm." [Online]. Available: https://www.npmjs.com/package/wikidata-sdk. [Accessed: 20-Apr-2018].
[35] J. Aliprand and Unicode Consortium., *The Unicode standard*. Addison-Wesley, 2003.
[36] Mozilla, "performance.now() - Web APIs | MDN." [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/API/Performance/now. [Accessed: 01-Jul-2018].
[37] S. Krug, "Rocket surgery made easy," *do-it Yours. Guid. to Find. fixing usability Probl. New Riders, Canada*, 2010.
[38] "Wikimedia Hackathon 2018 - MediaWiki." [Online]. Available: https://www.mediawiki.org/wiki/Wikimedia_Hackathon_2018. [Accessed: 01-Jul-2018].
[39] "Wikidata Query Service." [Online]. Available: https://query.wikidata.org/. [Accessed: 01-Jul-2018].

APPENDIX

A. GANTT DIAGRAM



B. UI Prototype Testing – Interview Template

Use case scenario: Looking for Pedro Dot (Template)

Hello and thank you for spending time with us. First of all, you need to know that this test does not have a good or bad response. We just want to know if everything in our app is fine or not. In fact, we need to know what's wrong, so don't worry about the results. If you have any questions during the test or after the test we will be happy to answer.

Tester information

Name:
Age:
Experience with mobile apps: High / Mid / Low

- **Describe what you see. What do you think it does?**
- **Insert text into the search box. We need to find the person "Pedro Dot Martinez". Do not click the search button.**
- **Look for a way to apply filters.**
- **How many filters do you find here? What do you think of this?**
- **Apply all of the filters.**
- **Press the search button (orange one)**
- **What do you find here, what do you think it is?**
- **Delete the Wikipedia filter.**
- **Go into the first result of the list (Pedro Dot Martinez)**
- **Scroll the page, explain me what do you see here.**
- **Try to open all the accordion fields. Scroll the page and let me know what do you think of this information.**

C. Prototype Testing Results

UI Prototype Testing: Problems and Solutions

Problem		Solution	
Default card image shown when it cannot retrieve a picture from Wikipedia is confusing, people think it is talking about Wikipedia, not related.		Change the default picture logo that represents an image, not Wikipedia.	
The subject of each Wikipedia card is not clear. User does not know what it refers to.	 	Add a header in each Wikipedia card subject (person, family name, occupation and historical context)	 
Users think that the text is not clear, it is too small to read.		Increase the text size despite it shows less text.	
When Wikipedia's "W" logo is used, users do not know what it means.		Change the "W" logo for the full "Wikipedia" logo.